CHAPTER 2

Literature Review

Abbes et.al (2010) in their paper a decentralized and fault-tolerant Desktop Grid system for distributed applications proposes a decentralized and fault-tolerant software system for the purpose of managing Desktop Grid resources. Its main design principle is to eliminate the need for a centralized server, therefore to remove the single point of failure and bottleneck of existing Desktop Grids. Instead, each node can play alternatively the role of client or server. Main contribution is to design the Pastry Grid protocol (based on Pastry) for Desktop Grid in order to support a wider class of applications, especially the distributed application with precedence between tasks in a parallel environment, to take care of failures.

Amoon (2012) in their Research **A** Fault Tolerant Scheduling System Based on Check pointing for Computational Grids says Job check pointing is one of the most common utilized techniques for providing fault tolerance in computational grids. The efficiency of check pointing depends on the choice of the checkpoint interval. Inappropriate check pointing interval can delay job execution. In this paper, a fault-tolerant job scheduling system based on check pointing technique is presented and evaluated. When scheduling a job, the system uses both average failure time and failure rate of grid resources combined with resources response time to generate scheduling decisions. The system uses the failure rate of the assigned resources to calculate the checkpoint interval for each job. Extensive simulation experiments are conducted to quantify the performance of the proposed system.

Anderson (2011) in E-science talks says that today personal computers are powerful but, most of the time, a large proportion of their computational power is left unused. A desktop grid takes this unused capacity, no matter what its location, and puts it to work solving scientific problems. With over 1 billion desktop computers in use, desktop grids can offer a low cost, readily available computing resource for scientists while allowing citizens across the world to contribute to scientific research. Together with grids and supercomputers, desktop grids can be a useful complement to the e-Infrastructure landscape.

Azeez and Haque (2012) in their research paper Resource Management in Grid Computing says as technology advances and the popularity and dependence on internet increases, advanced method of finding faster and cheaper solutions to computational problems are sought. It force lead to the development of a concept known as grid computing, which is a type of distributed computing, heterogeneous in nature because it makes an aggregated use of resources distributed over a large geographical area to solve problems usually complex ones on a larger scale. This paper provides a brief overview on grid computing and its resource management processes, important factors considered in resource management, comparison of different resource management processes and future outlook of grid computing and resource management.

Azeez et al(2011) in their paper grid computing with alchemi says middlewares allow submission of requests to execute a computation (called a Job) to the Grid, such that it can be run anywhere on the network. Therefore, grid Middlewares serve as an intermediary layer that allow a reliable and homogeneous access to resources managed locally with different syntax and access techniques. Within the context of availability of various Middlewares for Grid implementation with different features, this paper therefore focuses on various features that are peculiar to Alchemi by taking into consideration its Architecture, the Operating systems, software demand and limitation that are inherent from its usage.

Batista et.al(2007) in the paper self adjusting grid networks says a procedure called Traffic Engineering for grids for enabling grid networks to self-adjust to resource availability. The proposal is based on monitoring the State of resources and on task migration. It involves several layers of the Internet architecture.

Bheevgade, and Patrikar(2008) in their paper Implementation of Watch Dog Timer for Fault Tolerant Computing on Cluster Server says The difficulty in designing a Fault tolerant cluster system increases with the difficulties of various failures. The most imperative obsession is that the algorithm, which avoids a simple failure in a system, must tolerate the more severe failures. In this paper, they implemented the theory of watchdog timer

Choi et.al(2006) in their paper A Taxonomy of Desktop Grid Systems Focusing on Scheduling propose a new comprehensive taxonomy and survey of Desktop Grid in order to help understand the definition, architecture, model, and applications of Desktop Grid. Particularly, they focus on a taxonomy and survey of scheduling for Desktop Grid in this paper.

Chopra (2006) in his Research Proposed proposed backup manager concept. Backup manager uses the heart beating and replication based fault tolerant technique to monitor the central manager. In case of failure of the central manager, backup manager will take its control and avoids the grid to fail.

Dai et.al (2006) in their paper Reliability of grid service systems presents a model for the grid services and develops an approach to analyze the grid service reliability. Detailed algorithms to compute the grid service reliability are presented and a self-sensing technology is proposed for parameterization and monitoring. A numerical example is used for the purpose of illustration. Finally, the new model is compared with different types of conventional models, through which this model is verified more suitable for grid service reliability.

Das and Sarkar (2012) in their paper on fault tolerance of resources in computational grid says various heterogeneous resources of different administrative domain are virtually distributed through different network in computational grids. Thus any type of failure can occur at any point of time and job running in grid environment might fail. Hence fault tolerance is an important and challenging issue in grid computing as the dependability of individual grid resources may not be guaranteed. In order to make computational grids more effective and reliable fault tolerant system is necessary. The objective of their paper is to review different existing fault tolerance techniques applicable in grid computing. The paper presents state of the art of various fault tolerance technique and comparative study of the existing algorithms.

Dhir (2009) in their Paper Alchemi.NET Framework in Grid Computing says if the computer power is greater, the greater will be the accuracy in approximation i.e. close will be the approximation to the reality. The speed of the computer should be fast enough to do Teraflops, Pets FLOPS of the calculations in the small amount of time. The way to speed up the computation is to "parallize" it i.e. divide the work into pieces that can be worked on by separate processors at the same time. So instead of investing so much of money on Super Computer, we can built Computational grid having the power same that of the desktop computer. Computational Grid is used in number of applications like Weather Forecasting, Climate Change Prediction, Detection of Radio Signals emitted by intelligent Civilizations outside earth etc. In this paper Alchemi.NET framework is used for calculating the 100 digit places value after decimal of Pi (π) value.

Dhir et.al (2011) says in nimble@itccnogrid vs alchemi .net development of a novel toolkit namely Nimble@ITCEcnoGrid for calculating the pi() value up to 120 decimal places after decimal. Alchemi.NET is the other popular toolkit based on Windows Platform for doing the same purpose. There is a very interesting comparison between Nimble@ITCEcnoGrid toolkit and Alchemi toolkit as both runs on Windows Operating System.

Djilali and Herault (2004) in their research Toward Fault-Tolerant RPC for Internet Connected Desktop Grids with Volatile Node**s** says a new fault-tolerant RPC protocol associating an original combination of three-tier architecture, passive replication and message logging. They describe RPC-V, an implementation of the proposed protocol within the XtremWeb Desktop Grid middleware.

Duarte et.al (2006) in their paper Collaborative Fault Diagnosis in Grids through Automated Tests says current grid platforms must be augmented with a collaborative diagnosis mechanism. They propose for such mechanism to use automated tests to identify the root cause of a failure and propose the appropriate fix. They present a Java-based implementation of the proposed mechanism, which provides a simple and flexible framework that eases the development and maintenance of the automated test.

Ebeneze and Baskaran (2012) in their paper Fault Tolerant most Fitting Resource Scheduling Algorithm for Computational Grid proposes a replication technique to improve the fault tolerance of the fittest resource scheduling algorithm .Scheduling the task to the appropriate resource is a vital requirement in computational Grid. The fittest resource scheduling algorithm searches for the appropriate resource based on the job requirements, in contrary to the general scheduling algorithms where jobs are scheduled to the resources with best performance factor. They proposed a method to improve the fault tolerance of the fittest resource scheduling algorithm, by scheduling the job in coordination with job replication when the resource has low reliability.

Fulop (2008) in their research says the Complex computational and visualization algorithms require Large amounts of computational power. The computing power of a single desktop computer is insufficient for running such complex algorithms, and, traditionally, large parallel supercomputers or dedicated clusters were used for this job. A more convenient solution, which is becoming more and more popular, is based on the use of no dedicated desktop PCs in a Desktop Grid Computing environment. Harnessing idle CPU cycles, storage space and other resources of networked computers to work together on a particularly computational intensive application does this. Increasing power and communication bandwidth of desktop computers.

Jankowski et al (2008) in their work improving the fault-tolerance level within the GRID computing environment - integration with the low-level check pointing packages says the architecture and scenario of one of such proof-of-concept implementations. They presented integration of a low-level check pointing package with the Grid environment allows the Grid Resource Broker to recover user's jobs in case of failure.

Jhoney et.al (2010) in their United state patent Methods, apparatus and computer programs for automated problem solving in a distributed, collaborative environment claims that A set of software-implemented agents distributed across a network are used to identify agents and resources on the network that are capable of performing required tasks. The agents collaborate to generate a plan comprising a task allocation and task execution sequence in which tasks are allocated to selected ones of the agents and identified resources, and then the selected agents and resources execute the allocated tasks in accordance with the generated plan. Services (such as processing functions and knowledge) provided by hardware and software resources in a network are represented as `capabilities` of an associated agent. The functions implemented by each agent are also represented as capability.

Kandaswamy et.al (2008) in their paper Fault Tolerance and Recovery of Scientific Workflows on Computational Grids says describe the design and implementation of two mechanisms for fault-tolerance and recovery for complex scientific workflows on computational grids. They present algorithms for over-provisioning and migration, which are primary strategies for fault-tolerance. They consider application performance models, resource reliability models, network latency and bandwidth and queue wait times for batch-queues on compute resources for determining the correct fault-tolerance strategy.

Kondo et.al (2004) in their work Characterizing and Evaluating Desktop Grids construct a model of server equivalents for the desktop grids, which can be used to predict application performance. They present measurements of an enterprise desktop grid with over 220 hosts running the Entropia commercial desktop grid software. They utilize these measurements to characterize CPU availability and develop a performance model for desktop grid applications for various task granularities, showing that there is an optimal task size.

Kondo et.al (2009), in their research The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems says with the increasing functionality and complexity of distributed systems, resource failures are inevitable. While numerous models and algorithms for dealing with failures exist, the lack of public trace data sets and tools has prevented meaningful comparisons. To facilitate the design, validation, and comparison of fault-tolerant models and algorithms, we have created the Failure Trace Archive (FTA) as an online public repository of availability traces taken from diverse parallel and distributed systems. Their main contributions first, describe the design of the archive, in particular the rationale of the standard FTA format, and the design of a toolbox that facilitates automated analysis of trace data sets. Second, applying the toolbox, a uniform comparative analysis with statistics and models of failures in nine distributed systems. Third, they show how different interpretations of these data sets can result in different conclusions.

krauter et.al(2001) in their work A taxonomy and survey of grid resource management system for distributed system says an abstract model and comprehensive taxonomy for describing resource management architecture is developed. This is used to indentify approaches followed in the implementation in the existing resource management **.**System for very large scale network computing system known as grid. the results are used to identify architectural approaches & issues.

Latchoumy and Khader(2011) in their research paper survey of fault tolerance in grid computing says in grid computing, the probability of a failure is much greater than in traditional parallel computing. Therefore, the fault tolerance is an important property in order to achieve reliability, availability and QOS. In this paper, they give a survey on various fault tolerance techniques, fault management in different systems and related issues. A fault tolerance service deals with various types of resource failures, which include process failure, processor failure and network failures. This survey provides the related research results about fault tolerance in distinct functional areas of grid infrastructure and also gave the future directions about fault tolerance techniques, and it is a good reference for researcher.

Luther et.al (2005), in their paper Alchemi: A .NET-based Enterprise Grid Computing System present Alchemi, a .NET based Framework that provides the runtime machinery and programming environment required to construct enterprise/desktop grids and develop grid applications. It allows flexible application composition by supporting an object-oriented application programming model in addition to a file-based job model. Cross-platform support is provided via a web services interface and a flexible execution model supports dedicated and non-dedicated execution by grid nodes.

Luther et.al , in their research work Alchemi A .NET-based Grid Computing Framework and its Integration into Global Grids says Alchemi, a .NET-based grid computing framework that provides the runtime machinery and programming environment required to construct desktop grids and develop grid applications. It allows flexible application composition by supporting an object-oriented grid application programming model in addition to a grid job model.

Luther et.al in their paper Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework proposed a .NET-based Grid framework, called Alchemi. Alchemi provides the runtime machinery and programming environment required to construct enterprise grids and develop grid applications. It allows flexible application composition by supporting an object-oriented application programming model in addition to a file-based job model. Cross platform support is provided via a web services interface and a flexible execution model supports dedicated and non-dedicated execution by grid nodes.

Medeiros et.al (2003) in their paper Faults in Grids Why are they so bad and what can be done about it Computational Grids have the potential to become the main execution platform for high performance and distributed applications. Such systems are extremely complex and prone to failures. In this paper, they present a survey with the grid community on which several people shared their actual experience regarding fault treatment. The survey reveals that, nowadays, users have to be highly involved in diagnosing failures, that most failures are due to configuration problems, and that solutions for dealing with failures are mainly application-dependent. They identify two main reasons for this state of affairs. First, grid components that provide high-level abstractions when working do expose all gory details when broken. Since there are no appropriate mechanisms to deal with the complexity exposed (configuration, middleware, hardware and software issues).

Minhas et.al (2011) in their paper A novel cost-based framework for communication in computational grid using Anycast Routing says Computational grid can perform the computationally extensive jobs by utilizing the wide spread processing capabilities of volunteer processors. In order to utilize the wide spread resources, failure options cannot be ignored. In this paper they give the detailed implementation of fault tolerance techniques, and also propose a modified forwarding mechanism which forwards the request to the next hop from which more receivers are available, the main contribution of this paper is the implementations of fault tolerant techniques using any casting with modified forwarding mechanism and its analytical analysis for the computational grid.

Nadiminti et.al(2004) in their paper ExcelGrid A .NET Plug-in for Outsourcing Excel Spreadsheet Workload to Enterprise and Global Grids propose the use of grids as they allow us to harness geographically distributed computational resources for performing computations in parallel in order to speed up the execution. Paper describes a method to grid-enable Microsoft Excel, to execute spreadsheet computations on enterprise and global grids.

Nanadagopal and Uthariaraj (2010) in their paper fault tolerant scheduling for computational grid says reliability challenges arise because of the unreliable nature of grid infrastructure. Two major problems that are critical to the effective utilization of computational resources are efficient scheduling of jobs and providing fault tolerance in a reliable manner. This paper addresses these problems by combining the checkpoint replication based fault tolerance mechanism with Minimum Total Time to Release (MTTR) job scheduling algorithm.

Setiawan et.al (2004), in their paper Grid Crypt: High Performance Symmetric Key Cryptography using Enterprise Grids propose develop an application for symmetric key cryptography using enterprise grid middleware called Alchemi. An analysis and comparison of its performance is presented along with pointers to future work.

Smith (2001) in their paper A Framework for Control and Observation in Distributed Environments describe software framework for control and observation of resources, services, and applications that supports such uses and provide examples of how framework can be used.

Stelling et.al(1998) in their research paper A Fault Detection Service for Wide Area Distributed Computations says *a* variety of techniques exist for detecting and correcting faults, the implementation of these techniques in a particular context can be difficult. They proposed a fault detection service de-signed to be incorporated, in a modular fashion, into distributed computing systems, tools, or applications. This service uses well-known techniques based on un-reliable fault detectors to detect and report component failure, while allowing the user to trade timeliness of reporting against false positive rates. They describe the architecture of this service; report on experimental results that quantify its cost and accuracy.

Townend and Xu (2003) in their paper Fault Tolerance within a Grid Environment says begun to develop an approach for fault tolerance based on the idea of job replication, as anomalous results should be caught at the voting stage. This approach combines a replication-based fault tolerance approach with both dynamic prioritization and dynamic scheduling

Tuong (2000) in their research work Integrating Fault-Tolerance Techniques in Grid Applications says development of a framework for simplifying the construction of grid computational applications. The framework provides a generic extension mechanism for incorporating functionality into applications and consists of two models: (1) the reflective graph and event model, and (2), the exoevent notification model.

These models provide a platform for extending user applications with additional capabilities via composition.

Veeranjaneyulu and Srimathi (2012) in their paper fault tolerance in grid computing using wade Says Many grid applications will be running in environments where interaction faults are more commonly occur between diverse grid nodes. Resources may also be used outside of organizational boundaries; it becomes iteratively difficult to guarantee that a resource being used is not malicious one. Because of the diverse faults and failure conditions developing, deploying, and executing long running applications over the grid remains a challenge. Hence fault tolerance is a primary factor for grid computing. They proposed a prototype system is designed using agents to provide service replication, reactivation and avoids the single point of failure. The agents and the workflows are provided by a common software platform called WADE.

Versweyveld (2011) International science grid this week (ISGTW) in their article says today's personal computers are powerful. In fact every PC today is over 100 times more powerful than the Cray-1 machine that designed the stealth bomber in the 1970s. But, most of the time, a lot of their computational capacity goes to waste. Desktop grids can make use of this unused computational power. With more than one billion desktop computers in use, desktop grids can offer a low cost, readily available scientific resource." We are actually sitting on a huge source of computational power that is largely left unused in numbers of universities, research institutes, companies, home offices and households," said Leslie Versweyveld of the International Desktop Grid Federation (IDGF).

Vladoiu et.al (2009) in their paper Availability of Computational Resources for Desktop Grid Computing says the computing power of a single desktop computer is insufficient for running complex algorithms. A very convenient solution is based on the use of non-dedicated desktop PCs in a Desktop Grid Computing environment. The costs are highly distributed: every volunteer supports her resources (hardware, power, internet connections) while the benefited entity provides management infrastructures, (network bandwidth, servers, management services), receiving in exchange an enormous and otherwise unaffordable computing power. There has been little insight into the temporal structure of resource availability within an organizational setup. They present here observations regarding the availability of computing resources to be used in a desktop grid.

Weissman (1999) in their paper Fault Tolerant Computing on the Grid, What are My Options examined fault tolerance options for a common class of high-performance parallel applications, single-program-multiple-data (SPMD). Performance models for two fault tolerance methods, checkpoint-recovery and wide-area replication, have been developed. These models enable quantitative comparisons of the two methods as applied to SPMD applications.

Zhao et.al (2011) in their paper A Taxonomy of Peer-to-Peer Desktop Grid Paradigms says Desktop grid systems and applications have generated significant impacts on science and engineering. The emerging convergence of grid and peer-to-peer (P2P) computing technologies further opens new opportunities for enabling P2P Desktop Grid systems. Paper presents taxonomy for classifying P2P desktop grid implementation paradigms, aiming to summarize the state-of-the-art technologies and explore the current and potential solution space. To have a comprehensive taxonomy for P2P desktop grid paradigms, they investigate both computational and data grid systems.